A Session-aware DeepWalk Model for Session-based Recommendation

Kaiyuan Li
Beijing University of Posts and Telecommunications
Beijing, China
tsotfsk@bupt.edu.cn

Pengfei Wang
Beijing University of Posts and Telecommunications
Beijing, China
wangpengfei@bupt.edu.cn

Long Xia
Baidu Inc
Beijing, China
long.phil.xia@gmail.com

Abstract

Session-based recommendation is the task of recommending the next item a user might be interested in given partially known session information. Due to the diversity of interactions, only a part of the sessions contains product interactions. It is necessary to consider these sessions separately for better recommendations. In this paper, we describe our solution for SIGIR ecom 2021 Data Challenge. We first build two graphs based on the browsing and product interactions in sessions and apply the DeepWalk metric on these graphs to pre-train node embeddings respectively. For sessions having product interactions, we apply GRU on products to recommend users' next choices, while for sessions without product interactions, we directly analyze similarities between sessions and products for the recommendation. Our solution achieves an overall MRR@20 of 0.2628, which wins third place in the final leaderboard.

Keywords: DeepWalk, Session-based Recommendation, Data Challenge

1 Introduction

In this paper, we focus on the first task of SIGIR ecom 2021 Data Challenge[7], which is a session-based recommendation task aiming to predict the next interactions between shoppers and products based on the previous product interactions and searching queries within a session. In this task, pageview and product interactions in sessions are two important factors we need to consider. The different interactions can divide sessions into two parts: sessions containing product interactions, and sessions only containing page view interactions. We assume that these two types of sessions may indicate different user intentions. In addition, fusing these two different types of sessions into a unified framework is difficult, it is necessary to model them separately for a better recommendation.

To address this issue, in this paper, we design a session-aware DeepWalk model for recommendation. Specifically, we first build two graphs based on the browsing and product interactions in sessions respectively. Based on these two heterogeneous graphs, we then apply the DeepWalk metric on them to pre-train node embeddings. With the learned node representations, for sessions having product interactions, we apply GRU on products to recommend users' next choices, while for sessions without any product interactions, we directly analyze similarities between session and product nodes for recommendation. Our solution achieves an overall MRR@20 of 0.2628.

2 Motivation

In this section, we give a brief view of our motivation. We first make some analysis on the browsing interactions file. We found there exists duplicate user behaviors in sessions (i.e., user may click the same URL several times). To simplify, we remove all duplicated records by keeping the latest one. The statistics are shown in Table 1.

Table 1. The statistics of browsing data.

Property	Raw Data	Processed Data
# of sessions	5,408,429	5,408,429
# of products	60,182	58,711
# of all product interactions	10,489,498	4,182,749
# of pageview interactions	26,699,519	14,219,947
# of URLs	517,740	517,740

An interesting observation is that we find 66.2% of sessions only containing page view interactions. As cold start sessions, directly recommending products to these sessions is challenging. It is necessary to model these two types of sessions (sessions containing product interactions, and sessions only having page view interactions) separately for a better recommendation. Given this, we define two sub-tasks to address this issue: recommendation for sessions having product interactions, and recommendation for sessions only containing page view interactions.

3 Methodology

In this paper, we propose our session-aware DeepWalk model in detail, the structure of our model is shown in Fig. 1. In the following, we will give the design for two sub-tasks respectively.

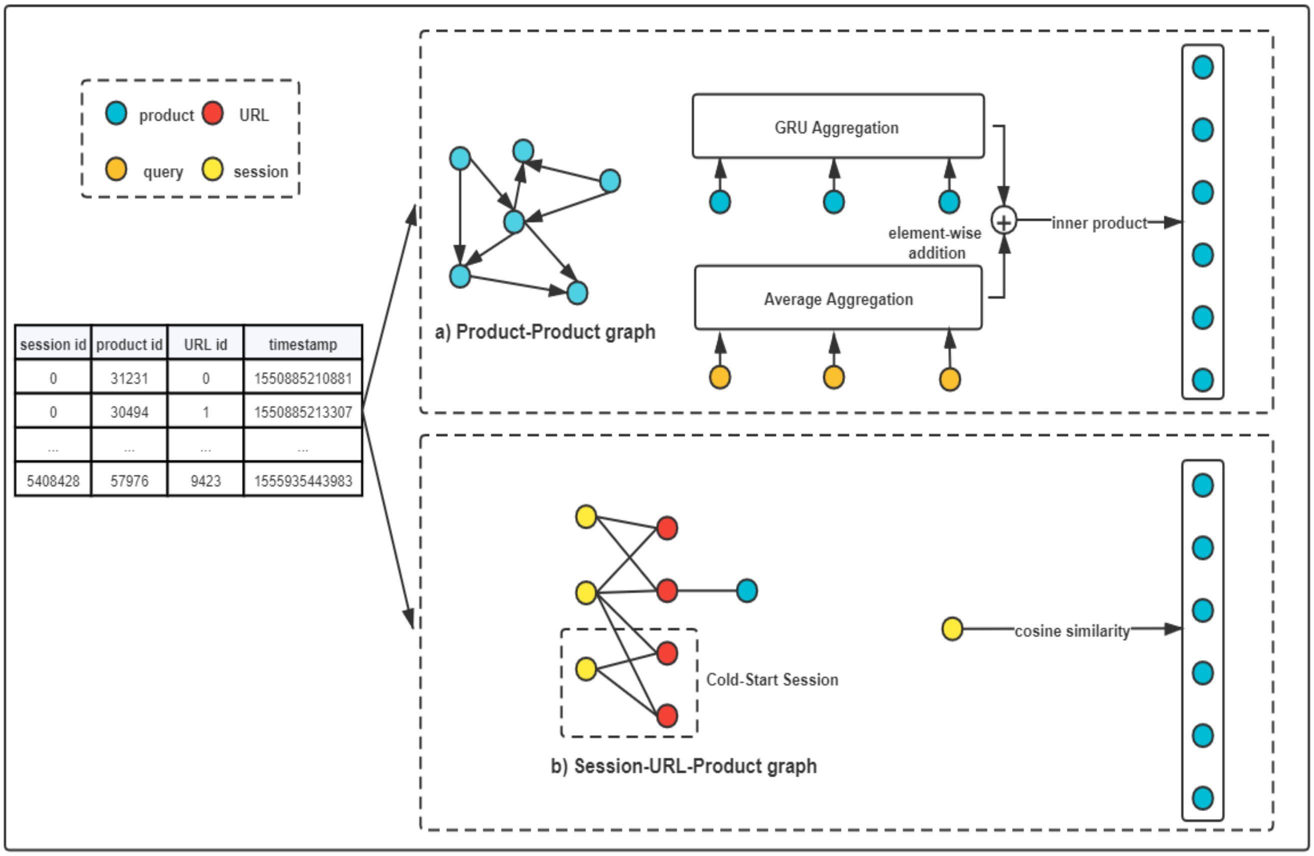

Figure 1. The overall architecture of our solution. For sessions having product interactions, we build a product-product graph and use DeepWalk to pre-train the node embeddings. After this, we use four strategies to initialize the GRU model and mix the sessions' query embedding to recommend products. For sessions only having page view interactions, we build a session-URL-product graph and use DeepWalk to pre-train the node embedding. After this, we use cosine to analyze similarities between session nodes and product nodes to recommend products.

3.1 Recommendation for sessions having product interactions

There are three main stages for the sub-task of recommending for sessions having product interactions, including the pre-training stage, recommendation stage, and recall stage.

3.1.1 Pre-training Stage on Product-Product Graph.

Considering product interactions can well reveal users' intentions, we construct a **p**roduct-**p**roduct graph by extracting all product interactions from sessions, where its nodes denote all interacted products in sessions, and each edge of it means these two corresponding products co-existed in the same session, and we use the occurrences of them as the weight of the edge.

After building the graph, we then use DeepWalk [6] to learn the representations for these nodes.

3.1.2 Recommendation Stage.

For each session s_p that has product interactions, we apply GRU to scan all its interacted products to obtain its sequential representation, denoted as $\vec{h}^{seq}_{s_p}$.

Considering that only using limited product interactions may weaken the recommendation performance, we further fuse search interactions into our model to capture users' intentions. Specifically, bases on the query embeddings offered, for each session s_p, we use average pooling to aggregate all its queries to represent its general interest, denoted as $\vec{h}^{general}_{s_p}$.

Based on $\vec{h}^{seq}_{s_p}$ and $\vec{h}^{general}_{s_p}$ that capture the sequential property and general interest of session s_p, the probability of choosing one product is written as :

$$P(v_i|s_p) = \frac{exp((\vec{h}^{seq}_{s_p} + \vec{h}^{general}_{s_p}) \cdot \vec{v}_i)}{\sum_{v_j \in I} exp(((\vec{h}^{seq}_{s_p} + \vec{h}^{general}_{s_p}) \cdot \vec{v}_j)} \tag{1}$$

where I denotes all products, $v_i \in I$ denotes i-th product, and $\vec{v_i}$ represents the embedding of v_i. We then maximize the log probability over the session data as follows:

$$L = \sum_{s_p} \sum_{v_i \in s_p} logP(v_i|s_p) \quad (2)$$

3.1.3 Multi-Channel Recall Stage. We consider four different product initialization strategies for recommendation:

- **Random Embedding**: We use random initialization as in [2] to initialize the product embeddings.
- **DeepWalk Embedding**: We utilize product embeddings learned in our pre-train stage for recommendation.
- **Text Embedding**: We use the text embedding offered to initialize the product embeddings.
- **Image Embedding**: We use the image embedding offered to initialize the product embeddings.

Based on the four initialization strategies mentioned above, we obtain four different models by changing product embedding in Eq.2. We use Adam[4] to train our model. With the learned parameters, given a session s_p, for each candidate product v_i, we calculate the probability $P(v_i|s_p)$ according to Eq.1 under four different product initialization strategies, and sum them as its final score. We then rank the products according to their scores and select the top 20 results as the final result.

3.2 Recommendation for Sessions Containing Only Page View Interactions

For the task of making recommendations for sessions only having page view interactions, we then use all browsing interaction data to construct a **s**ession-**u**rl-**p**roduct graph to learn their relations. Based on the learned embeddings of sessions and products, we then analyze their similarities for recommendation.

3.2.1 Pretraining on Session-URL-Product Graph. Considering this type of session has no product interactions, which brings difficulty to directly make a recommendation. To address this issue, we first construct a **s**ession-**u**rl-**p**roduct graph to analyze their relations, where its nodes contain all sessions, URLs, and products. For session nodes, we link them with their interacted URL and product nodes, for product nodes, we link them with their corresponding URL nodes. After building the graph, we then use DeepWalk to learn the representations for these nodes.

3.2.2 Embedding-based Retrieval. After getting all sessions and product representations, for each session having only page view interactions, we use cosine to analyze its similarities with all products. We rank the results and select the top 20 products as the final result.

4 Experiments

In this section, we further conduct a series of detailed experiments to analyze the effectiveness of our approach.

4.1 Setup

For each session, we sort its records according to the timestamp. We randomly select 3% of these sessions and hold out the last product to the validation set.

We use DeepWalk provided by DGL[8], and GRU4Rec provided by RecBole[9]. For hyperparameters of DeepWalk, in its sampling stage, we sample 10 paths for each node, and the length of each path is set to 80. In the training stage, the learning rate is 0.1, the number of negative samples is 5, the sliding window size is 2 and the embedding size is 128.

For all the sequential models, the maximum sequence length is 5, the batch size is 4096, the hidden size is 256, and the learning rate is 1e-4.

For the task of recommending sessions having product interactions, it needs about 4 hours for training, while for the task of recommending sessions containing only page view interactions, it will cost 6 hours. The inference time of our model is less than 5 minutes.

4.2 Performance Comparison with Different Product Initialization Strategies

Recall that we design four different product initialization strategies, and aggregate their results for recommendation. In this section, we analyze the performance of our model when utilizing different product initialization strategies separately. The result is shown in Table 2.

Table 2. Performance comparison under different initialization strategies. The best performance in each column is in bold font.

Strategy	MRR	F1	Coverage	PopBias
Random	0.2549	0.0655	0.4360	7.69e-5
Text	0.2561	0.0671	0.4350	7.73e-5
Image	0.2593	0.0666	0.4357	7.60e-5
DeepWalk	0.2607	0.0668	0.4363	**7.52e-5**
Ensemble	**0.2628**	**0.0678**	**0.4366**	7.59e-5

4.3 Performance Comparison with Baselines

We compare our proposed approach against the two following baselines:

- **Prod2Vec + KNN**[1]: a model that use Word2Vec[5] to pre-train the product embeddings, and then using the latest product in the session to calculate the cosine similarity with products.

- **GRU4Rec**[3]: a session-based recommendation, which utilizes GRU to capture sessions' long-term sequential behaviors.

For Prod2Vec+KNN, we use the code released by its authors, while GRU4Rec is implemented in PyTorch. The result is shown in Table 3.

Table 3. Performance comparison between the baselines and our model. The best performance in each column is in bold font.

Model	MRR	F1	Coverage	PopBias
Prod2Vec+KNN	0.1938	0.0465	0.2708	**3.28e-5**
GRU4Rec	0.2507	0.0643	0.4324	7.66e-5
Ours	**0.2628**	**0.0678**	**0.4366**	7.59e-5

4.4 Ablation Study

As in our model, we introduce two extra information for product recommendation, which are query embeddings, and page view interactions. In this section we conduct ablation study experiments to analyze their impacts respectively:

- **Without query embedding**: We remove all query embedding in the task of recommending for sessions having product interactions.
- **Without pageview sessions**: For sessions only containing page view interactions, we ignore all the impacts of these interactions, and directly recommend top 20 most popular products for each of them.

Table 4 shows the performance of these variation models.

Table 4. Ablation study on the test set.

Strategy	MRR
best model	0.**2607**
w/o query embedding	0.2565
w/o pageview sessions	0.2588

5 Conclusion

In this paper, we adopt a session-aware DeepWalk model for the session-based recommendation. We find that DeepWalk can well improve the downstream tasks. Moreover, for sessions only having page view interactions, DeepWalk can generate paths between these sessions and products, which will effectively alleviate the cold start problem.

References

[1] Federico Bianchi, Jacopo Tagliabue, Bingqing Yu, Luca Bigon, and Ciro Greco. 2020. Fantastic Embeddings and How to Align Them: Zero-Shot Inference in a Multi-Shop Scenario. arXiv:2007.14906 [cs.IR]

[2] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 9), Yee Whye Teh and Mike Titterington (Eds.). PMLR, Chia Laguna Resort, Sardinia, Italy, 249–256. http://proceedings.mlr.press/v9/glorot10a.html

[3] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. 2015. Session-based Recommendations with Recurrent Neural Networks. Computer ence (2015).

[4] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

[5] T. Mikolov, G. Corrado, C. Kai, and J. Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the International Conference on Learning Representations (ICLR 2013).

[6] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (New York, New York, USA) (KDD '14). Association for Computing Machinery, New York, NY, USA, 701–710. https://doi.org/10.1145/2623330.2623732

[7] Jacopo Tagliabue, Ciro Greco, Jean-Francis Roy, Bingqing Yu, Patrick John Chia, Federico Bianchi, and Giovanni Cassani. 2021. SIGIR 2021 E-Commerce Workshop Data Challenge. arXiv:2104.09423 [cs.IR]

[8] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. 2019. Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks. arXiv preprint arXiv:1909.01315 (2019).

[9] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Kaiyuan Li, Yushuo Chen, Yujie Lu, Hui Wang, Changxin Tian, Xingyu Pan, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2020. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. arXiv preprint arXiv:2011.01731 (2020).